Javascript : JQuery

Contenu

[Introduction 2](#_Toc522778652)

[Intégration dans une page web 3](#_Toc522778653)

[Appel des fichiers 3](#_Toc522778654)

[Point de démarrage 3](#_Toc522778655)

[Les sélecteurs 5](#_Toc522778656)

[Sélection d’une balise HTML 5](#_Toc522778657)

[Sélection d’une classe CSS 5](#_Toc522778658)

[Sélection d’un élément HTML avec identifiant 5](#_Toc522778659)

[Sélection multiple 6](#_Toc522778660)

[Sélection descendante 6](#_Toc522778661)

[Sélection d’éléments « commençant par » 7](#_Toc522778662)

[Sélection d’éléments « finissant par » 7](#_Toc522778663)

[Sélection d’un attribut 8](#_Toc522778664)

[Autres sélecteurs 8](#_Toc522778665)

[Manipuler l’objet sélectionné 8](#_Toc522778666)

[Fonction de manipulation des objets 9](#_Toc522778667)

[Manipuler les instructions CSS 9](#_Toc522778668)

[Modifier la hauteur/largeur d’un élément HTML 9](#_Toc522778669)

[Ajouter ou supprimer une classe 10](#_Toc522778670)

[Ajouter/modifier du contenu 10](#_Toc522778671)

[Effets Documentation Officielle 10](#_Toc522778672)

[Exemples 12](#_Toc522778673)

[Récupérer la valeur de l’attribut « id » de l’élément sélectionné 12](#_Toc522778674)

[Exercices 13](#_Toc522778675)

[1. Nombre magique : le retour 13](#_Toc522778676)

[2. Validation de formulaire 13](#_Toc522778677)

[3. Menu déroulant 13](#_Toc522778678)

# Introduction

[JQuery](https://jquery.com) est une librairie qui permet de simplifier l'usage du langage Javascript dans une page HTML en manipulant beaucoup plus facilement le DOM grâce à un système de sélecteurs.

JQuery peut permettre les points suivants :

* Manipuler des éléments du DOM (gestion des fenêtres)
* Manipuler des éléments HTML : ajouter, modifier, supprimer des balises, des attributs, du contenu ou encore masquer
* Manipuler le CSS : ajouter, modifier, supprimer des classes, modifier les couleurs etc.
* Animer des éléments : déplacer, rotation, dégradés de couleurs, transitions CSS etc.
* Faciliter les requêtes Ajax et la manipulation de données (au format Json notamment)

Son fonctionnement est le suivant : on exécute du code javascript sur un évènement via un sélecteur sur un évènement (clic, survol, soumission de formulaire etc.)

Exemple d’utilisation (fichier js externe) :

$(document).ready(function()

{

$("#bouton").click(function()

{

alert("Vous avez cliqué sur le bouton");

});

});

* $(document).ready(function() : initialise l’utilisation de JQuery au chargement de la page
* $("#bouton"): le sélecteur, qui cible l’élément passé en argument (ici une balise HTML dont l’attribut id est nommée bouton)
* .click : l’évènement, sur lequel on déclenchera le code contenu dans la fonction (ici affichage d’une boîte d’alerte). Le code Javascript de la fonction est celui que vous allez écrire : il va contenir à la fois du Javascript « normal » et du code spécifique à JQuery (fonctions etc.).

JQuery est apparu début 2006 et fut le premier outil de son genre. Il est devenu rapidement très populaire. Outre les fonctionnalités proposées, ses avantages sont sa compatibilité multi-navigateurs (en 2005, les navigateurs n’interprétaient pas tous le Javascript de la même façon), sa licence d’utilisation libre/gratuite et sa légèreté (quelques kilos-octets).

# Intégration dans une page web

Consulter [le site officiel de JQuery](http://jquery.com/download) pour connaître la dernière version de Jquery et mettre à jour le code du fichier à appeler (voir avec le formateur).

## Appel des fichiers

**Attention, si vous utilisez Bootstrap dans votre page HTML, vous chargez déjà jQuery, mais dans une version allégée (« slim ») !**

Comme tout fichier Javascript, il existe 2 possibilités :

* l’appel via un [CDN](http://fr.wikipedia.org/wiki/Content_delivery_network) :

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

Pour obtenir la dernière version de jQuery :

* Rendez-vous sur cette page
* Recherchez la section « jQuery Core - All 3.x Versions »
* Cliquez sur l’un des liens de la 1ère ligne (par exemple sur « minified ») et vous obtenez une fenêtre indiquant le code à intégrer (balise <script> complète).

Privilégiez une version dite « minifiée » (avec un .min dans le nom), nous reviendrons plus tard sur cette notion.

* le [téléchargement direct](https://jquery.com/download) (JQuery sera hébergé dans les répertoires de votre site) :

<script src="repertoire/jquery.min.js"></script>

JQuery peut aussi être installé via les nombreux gestionnaires de paquets qui existent désormais tels que *npm*, *bower* etc. Dans ce cas les fichiers de Jquery seront hébergés sur votre serveur.

## Point de démarrage

Après avoir chargé les fichiers Jquery dans une page HTML, il est nécessaire de préciser à cette page que l’on veut utiliser Jquery via l’instruction suivante à placer bien entendu dans une balise <script> ou un fichier JS externe :

$(document).ready(**function**()

{

// Code à exécuter

});

ready() est un évènement qui se produit en premier lorsque le navigateur a terminé de construire le [DOM](http://www.finalclap.com/tuto/cours-jquery-61/introduction-javascript-62/) (« parser » HTML) à partir du code source de la page (cela ne signifie pas que la page a terminé de se charger, on parle ici uniquement du HTML,  les fichiers de ressources CSS, images etc. ne sont pas forcément encore chargées en totalité).

ATTENTION : Un seul $(document).ready nécessaire par script Jquery.

Ensuite, on utilisera un sélecteur associé à un évènement (click, change, mouseover, blur…) pour cibler l’élément souhaité.

# Les sélecteurs

Les instructions JQuery commencent par la sélection d’un élément de la page web : balise HTML, attribut ou classe CSS que l'on veut manipuler.

Tout sélecteur commence par le signe $ (dollar) qui constitue un raccourci remplaçant les instructions en Javascript natif du type document.getElementBy… et document.querySelector…

$(selecteur).evenement

Sélection d’une balise HTML

Comme en CSS, il est possible de cibler toutes les balises d’un même type.

Tous les paragraphes :

$("p")

## Sélection d’une classe CSS

Sélection de tous éléments de la page auxquels est appliquée la classe passée en argument (ici la classe "rouge") :

$(".rouge")

## Sélection d’un élément HTML avec identifiant

Sélection d’une balise HTML nommé par un identifiant c’est-à-dire qui possède un attribut id*.* L’id est appelé par son nom précédé du signe # :

$("#bouton1")

Rappel : la valeur d’un attribut *id* doit être unique dans une page web : donc contrairement aux sélecteurs précédents ici un seul élément est pris en compte.

## Sélection multiple

Jquery permet de sélectionner différents éléments en une seule fois, là où il aurait fallu autant d’appels document.getElementBy… que d’éléments en Javascript natif. Les éléments doivent être séparés par une virgule :

// Plusieurs ID

$("#panneau1, #panneau2, #panneau3")

// Plusieurs classes

$(".rouge, .gras .noir")…

// Plusieurs balises HTML

$("p, div footer")…

Il est aussi possible de mixer balises, identifiants ou classes :

$("div, #panneau1, .rouge")…

## Sélection descendante

Le système de sélecteurs permet également d’atteindre les enfants d’un élément dans la hiérarchie D.O.M.

Il existe **plusieurs méthodes**, en fonction des besoins.

Exemple 1 : cibler tous les enfants d’un même type connu (ici, des <li>)

// HTML

<ul id="liste">

<li>Amiens</li>

<li>Paris</li>

<li>Lille</li>

</ul>

// Javascript/Jquery

// Sélectionne les enfants de l’élément #liste et leur met une bordure rouge)

$("#liste li").css("color", "1px solid red");

Exemple 2 : cibler tous les enfants de types différents :

// HTML

<div id="div1">

<h1>Je suis un enfant div1 !</h1>

<p>Je suis un fils de div1</p>

<div id="div2">Je descends de div1</div>

</div>

// Javascript/Jquery

// Sélectionne les enfants de l’élément #liste (met le texte en rouge)

$("#liste li").css("color", "1px solid red");

// Sélectionne les enfants de l’élément #div1, donc ici le h1, le paragraphe et l’élément #div2

## Sélection d’éléments « commençant par »

On peut sélectionner tous les éléments dont les valeurs des attributs id ou class « commencent par » :

// HTML

<ul>

<li id="listeA">Amiens</li>

<li id="listeB">Paris</li>

<li id="listeC">Lille</li>

</ul>

// JS

$("[id^='liste']")

## Sélection d’éléments « finissant par »

A contrario, on peut vouloir sélectionner tous les éléments qui « se terminent par »

// HTML

<ul>

<li id="Aliste">Amiens</li>

<li id="Bliste">Paris</li>

<li id="Cliste">Lille</li>

</ul>

// JS

$("[id$='liste']")

## Sélection d’un attribut

On peut cibler un attribut, cela se fait avec la méthode attr() :

// HTML

<a id="monLien" href="index.html" title="Page d'accueil">Accueil</a>

// JS

// Accède à l'attribut href de l'id 'monLien'

$("#monLien").attr("href");

## Autres sélecteurs

Il existe bien d’autres sélecteurs ou fonctions applicables aux sélecteurs, en voici [la liste](https://api.jquery.com/category/selectors/).

## Manipuler l’objet sélectionné

Comme dans Javascript tout est objet, l’utilisation d’un sélecteur renvoie l’élément ciblé en tant qu’objet D.O.M. avec l’ensemble des propriétés et méthodes qui lui sont attachées.

Cet objet est alors représenté par le mot-clé this qui en pratique sera lui-même utilisé comme sélecteur sous la forme $(this).

Exemple :

// HTML

<div id="p1">Le développeur logiciel est un scribe informatique. Sa mission : concevoir et tester des sites de contenu, d'e-commerce ou applications.</div>

// JS

$("#div1").mouseover(function() {

// Au passage de la souris, récupère le texte de l’élément #div1

// et le stocke dans la variable a

var a = $(**this**).text();

alert(a);

});

# Fonction de manipulation des objets

## Manipuler les instructions CSS

La méthode css() permet d’appliquer ou de modifier la valeur d'une propriété CSS :

css(nom de la propriété CSS, valeur souhaitée)

Exemple :

// Au passage de la souris, applique une bordure rouge sur l’élément #div1

$("#div1").mouseover(function()

{

$(this).css("border", "1px solid red");

});

Il est possible d’appliquer plusieurs propriétés à la fois; dans ce cas la syntaxe diffère : il faut ajouter des accolades dans les parenthèses et spécifier chaque couple de propriété/valeur séparé par une virgule :

$("#div1").mouseover(function()

{

$(this).css(**{**

"border" : "1px solid red",

"font-weight" : "bold",

"cursor" : "help"

**}**);

});

Enfin, la méthode css()permet de récupérer la valeur d’une propriété CSS :

$("#div1").mouseover(function()

{

var a = $(this).css("background-color);

// Affiche, par exemple, rgba(0, 0, 255) si fond rouge

alert(a);

});

* [Documentation officielle](http://api.jquery.com/css/)

## Modifier la hauteur/largeur d’un élément HTML

On obtient ou change la hauteur ou la largeur d'un élément grâce respectivement aux méthodes height() et width(), qui fonctionnent de la même manière :

$("#div1").click(function()

{

// Récupère la hauteur de #p1 et le stocke dans la variable h

var h = $(this).height();

// Si on passe une valeur, l’élément HTML est modifié en conséquence

// Ici, la hauteur sera modifié à 500px

$(this).height(500);

});

* Documentation : [height()](http://api.jquery.com/height/), [width()](http://api.jquery.com/width/)

## Ajouter ou supprimer une classe

Les méthodes addClass() et removeClass() ajoutent et suppriment respectivement une classe CSS à un élément HTML. La classe doit être définie préalablement dans la feuille de styles.

+++ TODO : exemples +++

* Documentation : [addClass()](http://api.jquery.com/addclass/), [removeClass()](http://api.jquery.com/removeclass/)

## 

## Ajouter/modifier du contenu

Jquery propose différentes fonctions pour ajouter, modifier ou supprimer du contenu HTML.

### Récupérer ou changer un texte : text()

// Au clic sur l’éléménet #lien

$("#lien").click(function()

{

// On récupère le texte de div1

var letexte = $("#div1").**text()**;

// Changer le texte (ici, copie le texte de div1 dans div2) :

$("#div2").**text(letexte)**;

});

La fonction text() peut contenir du HTML.

* [Documentation](http://api.jquery.com/text/)

### Ajouter ou remplacer du code HTML

+++ LIBELLE PARAGRAPHE A REVOIR, DIFFERENCE TEXT ET HTML +++

html(contenu)

Permet de modifier le contenu d'un élément HTML.

### Récupérer ou changer la valeur d’un input dans un formulaire : val()

val(name)

Obtient la valeur d'un élément INPUT

$("#champ").blur(function()

{

// Récupère la valeur siaise dans le champ input #champ

var r le texte (ici, copie le texte de div1 dans div2) :

$("#div2").**text(letexte)**;

});

## Effets [Documentation Officielle](http://api.jquery.com/category/css/)

Afficher, cacher ou inverser un élément : fonctions show(), hide(), toggle() :

// Affiche ou cache l’élément div1

$('#div1').show();

// Modifie le contenu (innerHTML) de l’élément id=”div1”

$('#div1').html('coucou<br>c\'est moi');

fadeIn(), fadeOut(), fadeToggle()

Afficher, cacher ou inverser un élément (avec l'effet fade)

slideDown(), slideUp(), slideToggle()

Afficher, cacher ou inverser un élément (avec l'effet slide)

animate(propriétés, délai)

Déclenche une animation sur les propriétés spécifiées pendant le délai indiqué

$("#div1").animate({ opacity: 0.25, left: "+=50" }, 5000);

+++ FORMULAIRES +++

e.PreventDefault()

# Exemples

## Récupérer la valeur de l’attribut « id » de l’élément sélectionné

Il est important de comprendre que le sélecteur devient un objet Javascript ; c’est cet objet qui est manipulé (via la variable this notamment) et non pas, comme on pourrait le penser, l’identifiant lui-même.

Par conséquent, si on souhaite récupérer la valeur de l’identifiant, il faut utiliser le mot-clé this combiné avec la méthode attr() :

$("#div1").click(function()

{

// Récupère la valeur de l’attribut id de l’élément #div1

var identifiant = **$(this).attr("id")**;

// Affiche : div1

alert(identifiant);

});

## Contrôler un formulaire

Essayez le code suivant : able this notamment) et non pas, comme on pourrait le penser, l’identifiant lui-même.

Par conséquent, si on souhaite récupérer la valeur de l’identifiant, il faut utiliser le mot-clé this combiné avec la méthode attr() :

$("#div1").click(function()

{

// Récupère la valeur de l’attribut id de l’élément #div1

var identifiant = **$(this).attr("id")**;

// Affiche : div1

alert(identifiant);

});

<form>

<p><label for="nom">Nom</label> <input type="text" name="nom" id="nom"></p>

<p><label for="nom">Prénom</label> <input type="text" name="prenom" id="prenom"></p>

<p><input type="submit" id="envoi" value="Envoyer"></p>

# Exercices

## Validation de formulaire

Reprenez l’exercice de validation du formulaire « sondage stagiaires », réalisez les contrôles à l’aide de JQuery.